VĚDECKÉ SPISY VYSOKÉHO UČENÍ TECHNICKÉHO V BRNĚ

Edice Habilitační a inaugurační spisy, sv. 606
ISSN 1213-418X

Dan Komosný

CYBERGEOGRAPHY AND CYBERSECURITY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

doc. Ing. Dan Komosný, Ph.D.

CYBERGEOGRAPHY AND CYBERSECURITY

KYBERGEOGRAFIE A KYBERBEZPEČNOST

TEZE PŘEDNÁŠKY
K PROFESORSKÉMU JMENOVACÍMU ŘÍZENÍ
V OBORU
TELEINFORMATIKA

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
NAKLADATELSTVÍ VUTIUM

BRNO 2018

KEYWORDS

Internet, location, IP address, security

KLÍČOVÁ SLOVA

Internet, poloha, IP adresa, bezpečnost

ISBN 978-80-214-5662-4
ISSN 1213-418X

CONTENTS

CURRICULUM VITAE

Assoc. Prof. Dan Komosny, Ph.D.

*1976, Czech Republic

Brno University of Technology
Faculty of Electrical Engineering and Communication
Department of Telecommunications
Technicka 12, 616 00 BRNO, Czech Republic

Phone: +420 54114 6973
Email: komosny@vutbr.cz

Current position

Associate Professor, Brno University of Technology, 2009–present

Senior Researcher, SIX Research Centre, 2012–present

Previous positions

Technical consultant, SIX Research Centre, 2010–2012

Assistant Professor, Brno University of Technology, 2003–2009

Computer programmer-analyst, Herman Electronics, 1998–2003

Education

Ph.D. in Teleinformatics, Brno University of Technology, 2003

Ing. in Electronics and Communication, Brno University of Technology, 2000

Research

Data networks, Cyber geography, Cyber security

Course lecturer

Network operating systems (fundamental), 2004–present

Cisco networking academy (optional), 2007–present

Awards

Award by rector of Brno University of Technology for research result

Five best paper awards at international conferences

Academic recognition

Supervisor of 6 doctoral students (Ph.D.)

Member of 28 doctoral exam and final doctoral committees (Ph.D.)

Chairman/member of 48 state exam committees

Teaching at 8 universities abroad

Research recognition

20 articles in recognized journals with impact factor by Web of Science (9 as first author)

Selected recognized journals – IEEE Access, Journal of Network and Computer Applications, Computer Standards & Interfaces, Telecommunication Systems

Project leader

12 projects funded by Czech Ministry of Education, Youth and Sports (MSMT), Czech Science Foundation (GACR), Czech Academy of Sciences (AVCR), and CESNET

Other

Reviewer of doctoral theses, projects, journal and conference papers (more than 150); member of journals and conferences abroad, invited research lectures abroad

1 INTRODUCTION

Cybergeography is a scientific field dealing with geographical mapping of the Internet virtual space. It is an interdisciplinary research of computer networking and physical geography. The virtual space of the Internet and its data covers various aspects, such as device interconnections, data flows, routing policies, user behaviour and logical topology layouts. All these aspects may be related to geographical locations. The research area of Cybergeography is important for gaining new fundamental knowledge about the nature of the Internet and on-line user behaviour. It is also important for the related applied research, such as Cybersecurity.

This lecture specifically deals with the spatial location of Internet devices, both end and intermediate. It focuses on device-independent location where the location of a device is estimated remotely by its known IP (Internet Protocol) address and not locally by the device itself, such as using an in-built GPS module. Therefore, the lecture focuses on the general scenario provided by device-independent spatial location (also known as IP geolocation) that

- does not use GPS or other global positioning systems,
- does not use local terrestrial radio-based location systems, such as triangulation in WiFi or mobile cellular networks,
- does not use location information entered by users.

The location is estimated based on the knowledge of a device's IP address (further referred to as the target), which may be obtained by various means (such as via log files storing previous device access to a service). The known spatial location of Internet devices is used for a great number of location-aware services and applications, including web content and social network personalization [1], on-line user behaviour analysis [2] (including visitor maps for websites), and load balancing by redirecting users to geographically close data/resource replicas [3]. Cybersecurity related applications may be the detection of identity theft [4], detection of credit card fraud [5], verification of server authenticity [6], and avoidance of risky geographical areas during Internet communication [7].

In contrast, the device-dependent location involves specific communication with the device located. The location is typically obtained from the devices' in-built GPS or WiFi triangulation. Typically, the device user has to agree with sharing their location. This may happen when an application/service is being installed and a user accepts the legal agreements. Applications of Internet device-dependent location vary a lot. Typical examples are finding nearby points of interest, finding nearest social media contacts, traffic reports, and municipal transport schedules.

The general methods used for device-independent location are summarized in table 1. The first three are used by the general public. These methods can be accessed freely or through a paid subscription to a commercial location service. The last method is typically dedicated for legally authorized entities only (police and other legal authorities).

The first method – registry databases – is based on looking up contact information stored in IP space allocation registries. IP address blocks allocated to specific organizational entities are recorded along with the entity contact/postal information. This data are used as the location information for the targets within these IP address ranges.

The second method – geolocation databases – deals with searching the dedicated databases. A geolocation database maintains blocks of continuous IP addresses. Each block has a location assigned. The assigned locations are obtained through several sources. A location of the target

Table 1: Overview of device-independent spatial location.

How	Accuracy	Used by
Registry databases	Very low	Non-device user (everybody)
Geolocation databases	Moderate	Non-device user (everybody)
Latency measurements	Low	Non-device user (everybody)
ISP private data	High	Non-device user (authorized)

is estimated by searching the database for the corresponding block of IPs and if a match is found, the geographical position stored for the block is returned [8].

The third method – latency measurements – is based on capturing and analysing the communication data. Typically, communication latency is measured from a set of servers to the target. These servers are geographical landmarks with known locations. The latencies are converted to a geographical region including the target using various techniques, such as distance multilateration [9].

The last method – ISP private data – is based on private records of ISP (Internet Service Providers). An ISP leases IP addresses to the devices. A country law specifies what information has to be recorded and how long it has to be maintained by ISPs. Internet connectivity subscribers provide their details including postal addresses in the billing contracts. The location of the target can be tracked down by linking this information. The legal details may vary across countries. The police and justice services are usually authorised to be given the device's location upon a formal request.

The accuracy of device-independent location is generally low, usually in a range of tens of kilometres. The accuracy of the ISP internal location information varies according to the records kept.

There are other minor methods of device-independent location, such as data-mining of web pages and other Internet resources for spatial information [10]. Another method is based on an enhancement of the DNS (Domain Name System) service called DNS LOC [11]. This service provides the geographical location for a domain name. The disadvantage of this solution is in a poor coverage of Internet address space and it is not widely used [12].

Different methods are used for device-dependent location. These methods are usually based on global positioning systems (GPS), measuring radio signal strength (RSSI), time of arrival (TOA), and angle of arrival (AOA) [13]. The accuracy of device-dependent location is higher than device-independent. Some principles are shared with device-independent location, such as multilateration.

The methods based on publicly available data are considered in this lecture. These may be further categorized as:

- Internet address space and domain registration databases,
- extension of domain name system,
- dedicated geolocation databases,
- measurement-based principles.

2 REGISTRY DATABASES AND DOMAIN NAMES

The basic approach based on registry databases is included to demonstrate the problems of device-independent spatial location. A trivial approach for mapping targets to their geographical location is to use data provided by the IP address space registries. The global use of Internet address space is controlled by IANA (Internet Assignment Numbers Authority). IANA allocates the major segments of IP addresses to five regional Internet registries (RIRs) – AFRINIC (Africa), APNIC (Asia/Pacific), ARIN (North America), LACNIC (Latin America), and RIPE NCC (Europe, the Middle East, and Central Asia). The regional registries further allocate IP address segments to ISPs. Such allocation can be direct or through two types of intermediary entities – national internet registry (NIR) and local internet registry (LIR). Table 2 lists the entities involved in the IP space allocation along with the minimum network prefixes that can be allocated [14] (network prefix divides an IP address into the network section and host section; network section defines the particular network).

Table 2: Entities involved in IP address space allocation with minimum network prefix size.

-	IANA	RIR	NIR	LIR/ISP	End user
IPv4	-	/8	/20 - /22	/20 - /22	vary
IPv6	-	/12	/32	/32	/48 - /64

The allocated IP address segments are stored in a database managed by a regional registry. Along with the allocation records, the registries maintain contact information of the organizations with assigned IP address ranges. The stored contact information provides a way to locate a target in some extent.

However, there are no official rules for filling the contact information by the organizations and thus the provided location can lead to wrong results. The next major concern is that the IP addresses falling into one allocation segment can be distributed over a large geographical area depending on the type and size of the organization. Examples include ISPs that operate at the national level or organizations with branches at different locations. In the case of small and local organizations, IP addresses may be used within a single city and thus giving a sufficient city-level accuracy. Further information about the location accuracy can be found in [15].

The stored contact information in the registry databases can be accessed by querying a particular IP address as demonstrated in listing 1. The listing shows the `whois` command, which is commonly used for this purpose. The reply comes from the RIPE NCC registry. Another possibility is to request location information for registered domain names at national level; this example is shown in figure 2. The result comes from the registry CZ.NIC of the `.cz` domain.

```
[]$ whois 147.229.147.X
...
address:        Brno University of Technology
address:        Antoninska 1
address:        601 90 Brno
address:        The Czech Republic
```

Listing 1: Basic method – location information stored in registry database (RIPE NCC).

```
[]$ whois vutbr.cz
...
address:       Antoninska 548/1
address:       Brno
address:       601 90
address:       CZ
```

Listing 2: Basic method – location information stored for domain name (CZ.NIC).

Both listings show examples with a specific location. However, as noted above, this location is not accurate as it may cover a number of Internet devices over a large geographical area. Also, there are no specific rules for the contact information to be stored in registries, and the stored locations are not verified for their validity. Furthermore, there are no rules for geographical naming despite some standardization efforts [16]. An exception is that some ISPs use internal geographical naming schemes for their intermediate devices and such information can be used as source data for IP geolocation [17].

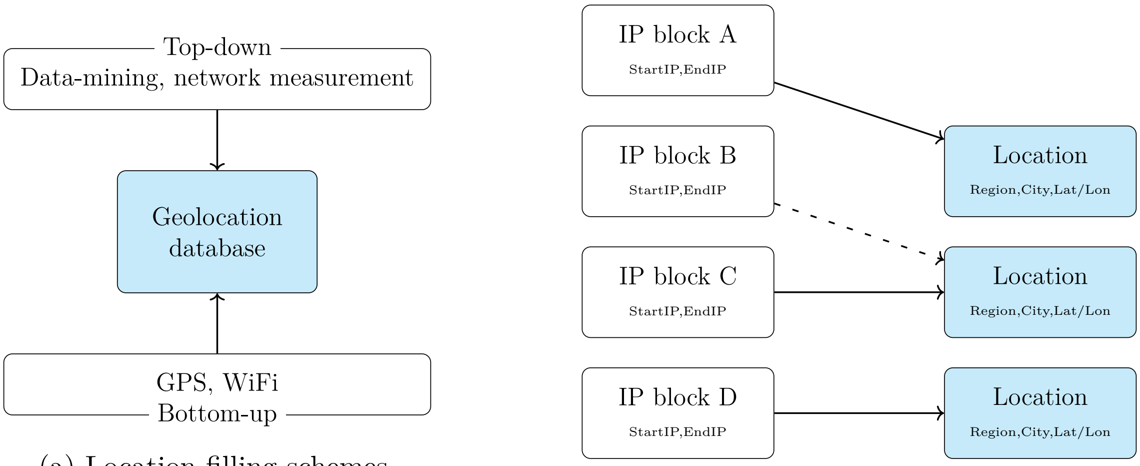
An enhancement was proposed to the domain name system [18, 19, 11] as a result of the poor accuracy of the data in registry databases. The proposed enhancement defines a new LOC record which stores latitude, longitude, and altitude for domain names. An example DNS LOC-based location is shown in listing 3 [12]. The listing shows an output of the `dig` command used to query the domain name servers. However, this enhancement gives poor location efficiency (i.e. high number of unresolved locations) [12].

```
[]$ dig loc alaska.net
...
;; ANSWER SECTION:
alaska.net. 600 IN LOC 61 11 0.000 N 149 50 0.000 W 10.00m
```

Listing 3: Location information stored in DNS LOC record.

3 DEDICATED GEOLOCATION DATABASES

A current approach is to use dedicated geolocation databases that extend geographical data from IP address space allocation registries. A geolocation database is structured into blocks of adjacent IP addresses and stores a geographical location for each block. The blocks may have varying size. The IP address blocks are typically smaller (have a larger network prefix) than the allocation segments stored in the registries and thus provide a better location resolution. There are different sources of locations associated with the blocks. This information is obtained using two schemes – top-down and bottom-up – as shown in figure 1a. The top-down scheme uses location information available through Internet resources, such as measuring the network and crawling the web [10]. The bottom-up scheme uses locations collected by external resources, such as GPS or WiFi network scanning.



(a) Location filling schemes.

(b) Organization of IP address blocks.

Figure 1: Dedicated geolocation databases.

An example construction of a geolocation database is shown in figure 1b with real data items shown in listings 4 and 5. The listings are based on the free-to-use database GeoLite2-City by MaxMind [20]. The database uses the CSV format. Both listings have been simplified for clarity. The first listing shows a sample content of the first file, where the blocks of continuous IP addresses are defined. A block is delimited by the first IP address and its size, which is specified by the network mask. The next items are the geographical name ID, country ID, if anonymous proxy use is considered, whether a satellite provider is present, postal code, the bound location information in the form of latitude and longitude, and accuracy radius in kilometres. As noted above, IP blocks may vary in their size. The listing shows three block sizes – /25,/26, and /27. A higher network mask means a lower number of host IP addresses assigned to a location. The second listing shows additional location information assigned to the block of continuous IP addresses. It starts with the geographical name ID, following is language used for place names, continent, country, subdivision level 1 (region), subdivision level 2 (district), and city.

```
89.102.17.0/25,3079332,3077311,0,0,664 01,49.2500,16.6667,5
81.19.4.64/26,3078610,3077311,0,0,614 00,49.2000,16.6333,10
185.137.125.32/27,3078610,3077311,0,0,627 00,49.2000,16.6333,10
```

Listing 4: Blocks of continuous IP addresses with assigned location information.

```
3079332,en,Europe,Czechia,South Moravian,Brno-Venkov,Bilovice nad Svitavou
3078610,en,Europe,Czechia,South Moravian,Mesto Brno,Brno
```

Listing 5: Location information for blocks of continuous IP addresses.

Both listings show that more IP blocks can share one assigned location (geographical name id 3078610, Brno). Such locations are typically the geographical centre of a country or centre of the capital city. These places are used when a more specific location for the IP address block is not available (this is also demonstrated in table 3). Other possible shared location is the geographical centre of a city, which is used when only the city name is known for an IP address block.

Geolocation databases can be accessed remotely or locally. Remote access is preferred for a low number of location lookups, as each lookup generates traffic and is delayed. In this case, the location service provider maintains the database to be updated. Usually, the number of location queries is recorded and the service provider charges the subscribers based on these numbers. An example of local database access for an IP address with the estimated result accuracy radius of 5 km is shown in listing 5. The location was searched in the previously downloaded database GeoLite2-City by MaxMind [20].

```
[]$ mmdblookup --file GeoLite2-City.mmdb --ip 147.229.147.X
city: Brno; country: Czechia; latitude: 49.200000; longitude: 16.633300;
    postal code: 614 00; subdivision: South Moravian; accuracy radius: 5
```

Listing 6: Example location using local geolocation database.

Geolocation databases are known for the large diversity of location accuracy. An example in figure 2 shows an error range from few kilometres to 200 km for a target in the Czech Republic. The locations plotted were obtained from seven free-to-use location databases:

- GeoLite2 City by MaxMind (`dev.maxmind.com/geoip/geoip2/geolite2/`), referred to as GeoLite2.
- IP address to city (low resolution) by DB-IP (`db-ip.com/db`), referred to as IPtoCity.
- DB11.LITE by IP2Location (`lite.ip2location.com`).
- Lite Free by IPligence (`www.ipligence.com/free-ip-database`).
- hostip (`www.hostip.info`).
- freegeoip (`www.freegeoip.net`).
- software77 (`software77.net/geo-ip`).

The map shows that the estimated locations are sometimes far apart. Some pointed to the capital city of the country (Prague) – IPtoCity and DB11.LITE. Other databases estimated only the correct country – Lite Free and software77. In this case, the geographical centre of the country was used. The other two databases estimated the correct region, but the wrong city – GeoLite2 and freegeoip. Finally, the hostip database returned a location outside the correct country.

The location errors are summarized in table 3. The correct location was at latitude 48.97 and longitude 16.61. The location errors varied from 7 km (GeoLite2) to 201 km (DB11.LITE).

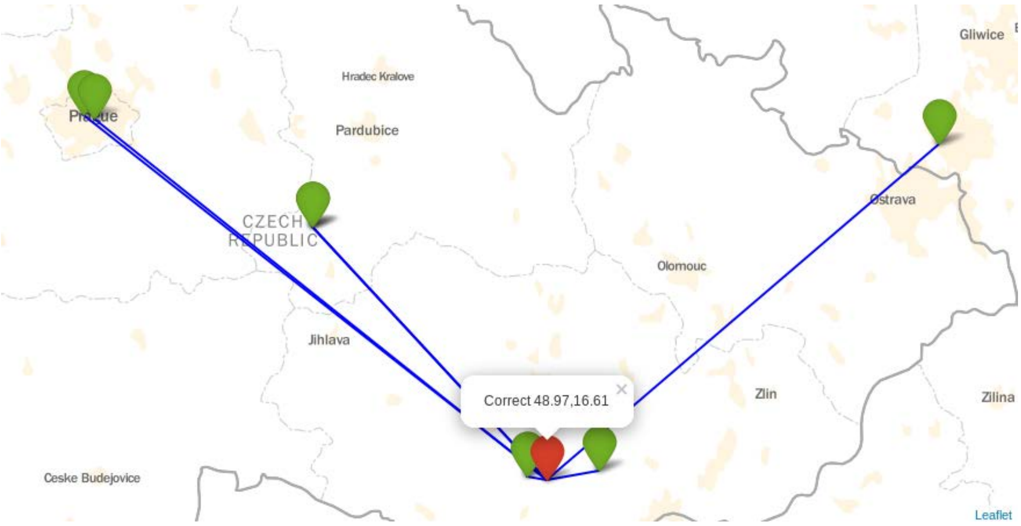


Figure 2: Example of error variations obtained from seven free-to-use IP geolocation databases.

Some databases returned directly the coordinates of the estimated location. In the cases when the coordinates were not returned directly, they were derived as the geographical centre of the returned city/region/country, respectively. These values are marked with '*'. The databases also vary in what information they provide. Table 4 gives an overview of the possible location information provided [21, 22, 23].

It is worth mentioning that the free-to-use databases have a lower accuracy compared to commercial databases [15]. The free-to-use databases are natively included into many UNIX and Linux operating systems. For example, the `geoiplookup` and `mmdblookup` (works with newer database format) commands may be used to access the pre-installed or downloaded databases by MaxMind. The claimed location accuracy by eight major commercial providers of the geolocation databases is summarized in table 5. The databases typically return geographical coordinates (latitude and longitude), country, region, and city. An exception is the IPligence product 'Pro' that does not return a region. The Skyhook database 'Hyperlocal IP Pro' differs from the others by returning a region and city only when the estimation reaches a certain level of location correctness.

MaxMind published the accuracy data for 23 countries [24]. For the purpose of comparison at the country level, the maximum location error of 250 km was used to evaluate a result as correct. 4 % of the location queries were reported to be resolved incorrectly, and 12 % of location queries were reported to be unresolved at the country level. For the city level, 48 % of

Table 3: Accuracy of locations obtained from seven free-to-use geolocation databases.

Vendor/Database	Lat, Lon	Error [km]	Place note
MaxMind/GeoLite2	48.98, 16.52	7	Correct region
DB-IP/IPtoCity	50.08, 14.47 *	190	Capital city
IP2Loc./DB11.LITE	50.09, 14.42	201	Capital city
IPligence/Lite Free	49.75, 15.5 *	118	Centre of country
hostip	50.0, 18.47	177	Wrong country
freegeoip	49.0, 16.86	19	Correct region
software77	49.75, 15.5 *	118	Centre of country

Table 4: Overview of possible information stored in geolocation databases.

Geographical	Networking	Demographical	Other
Country	Domain	Population density	Confidence factor
Region	ISP	Average income	Accuracy radius
City	AS Number	Organization type	Proxy, VPN
Lat,Lon	Connection speed	-	Fixed/mobile
Postal code	-	-	Place type (e.g. airport, hotel)

the location queries gave an incorrect city, and 11 % of the location queries were unresolved. DB-IP did not publish any data on location accuracy, only the range of IP address space covered. IP2Location published comprehensive location accuracy data for 250 countries [25]. The data covered only the city coverage (the country level was not included). For the purpose of comparison at the city level, the maximum error of 50 miles was used. Neustar did not publish any data on the accuracy of their geolocation services. However, the accuracy of the databases was evaluated by an external organization PricewaterhouseCoopers [26]. The result was 99.9 % accuracy for the country level. Neustar claimed to cover the entire IP address space. IPAddressLabs did not provide any location accuracy information about their products. The only information provided was that it covered the whole IPv4 address space. Geobytes provided some basic data about their accuracy [27]. It claimed to resolve 98 % of IP addresses with an accuracy of 97 % at the country level. Other information published was that 80 % of location queries resulted within the maximum error of 100 km and 75 % of locations were within the maximum location error of 50 km. IPligence did not publish any accuracy related information. The same held for Skyhook.

Comparing location accuracy is not straightforward due to the fact that researchers use different evaluation techniques. An example is using specific distance thresholds for the city-level accuracy, which can range from 40 to 100 km. The same problem holds for the database vendors when presenting their results. Therefore, a different approach was used to compare the location accuracy – cumulative probabilities for targets located within a maximal error of 50, 100, 150, and 250 km (these error ranges are used in table 6).

In [28] the authors stated that the location accuracy evaluation was difficult due to the use of different groundtruth datasets. Their established groundtruth was based on an algorithm which grouped IP addresses into virtual points of presence (PoPs). The algorithm discovered sets of routers at the same location. They used latency measurements and topology discovery

Table 5: Claimed accuracy of major geolocation databases.

Vendor/database	Country [%]	City [%]	IPv4	IPv6
MaxMind/GeoIP2 Precision	84	40	100 %	YES – NA
DB-IP/IP address to location + ISP	NA	NA	7 mil.	YES – 586,718
IP2Location/DB24	NA	77	14 mil.	YES – NA
Neustar/where	99.9	NA	100 %	YES – 100 %
IPAddressLabs/professional edition	NA	NA	100 %	NO
Geobytes/Geo IP Location	97	75	98 %	NO
IPligence/IPligence Pro	NA	NA	NA	NO
Skyhook/Hyperlocal IP	NA	NA	NA	NO

for this purpose. Therefore, the location accuracy of the databases depended on the exactness of the established PoPs. Six major geolocation databases were evaluated: MaxMind, IP2Location, IPligence, HostIP, Netaculity, and Geobytes. The results are summarized in table 6. The values were estimated from the cumulative probability functions presented in [28] for the selected maximum error ranges.

Table 6: Cumulative percentage of estimated locations within maximum location error [km].

Vendor	< 50	< 100	< 150	< 250
MaxMind	68	73	76	78
IP2Location	62	65	66	68
IPligence	73	75	76	78
HostIP	37	39	42	45
Netaculity	45	49	50	54
Geobytes	33	35	40	45

4 COMMUNICATION LATENCY PRINCIPLES

Communication latency consists of a set of partial contributors caused by different factors. The main contributors are the length and speed of links on the path, number of intermediate devices and their actual load. Communication latency d at the instant k for N links on the path may be described as introduced in [29, 30]

$$d(k) = \sum_{i=1}^{N} \left(\frac{p_i}{\mathrm{C}} + d_{i-1}^{\mathrm{RD}} + d_{i-1}^{\mathrm{RS}}(k) + \frac{M}{b_i} \right) = \sum_{i=1}^{N} \left(\frac{p_i}{\mathrm{C}} + d_{i-1}^{\mathrm{RD}} + \frac{M}{b_i} \right) + \sum_{i=1}^{N} d_{i-1}^{\mathrm{RS}}(k), \tag{1}$$

where p_i is the length of the ith link on the path, d_i^{RD} is the deterministic routing delay of the ith router on the path, d_i^{RS} is the variable stochastic routing delay caused by the actual load of the ith router on the path, $d_0^{\mathrm{RD}} = 0$ and $d_0^{\mathrm{RS}}(k) = 0$ is the void router assigned to the first link, M is the length of the packet transmitted, b_i is the bandwidth of the ith link, and the constant C is the speed of light in a vacuum. The part $\frac{p_i}{\mathrm{C}}$ also covers delay caused by switches and possible other devices on the link. Equation (1) is further divided into two parts d^{D} and $d^{\mathrm{S}}(k)$, where d^{D} is the independent deterministic part of time and $d^{\mathrm{S}}(k)$ is the dependent stochastic part. The difference between the deterministic and stochastic parts is shown in figure 3. The figure plots a histogram of the delay values for communication between two end hosts over a specific distance [31]. The measurement was carried out in the PlanetLab network, which is described later. When considering end-to-end communication latency, typically measured as RTT (Round Trip Time) using the ICMP (Internet Control Message Protocol) request-reply messages, the factor of the end hosts (requesting and replying) is also included. Therefore, the deterministic one-way part d^{D} is modified as

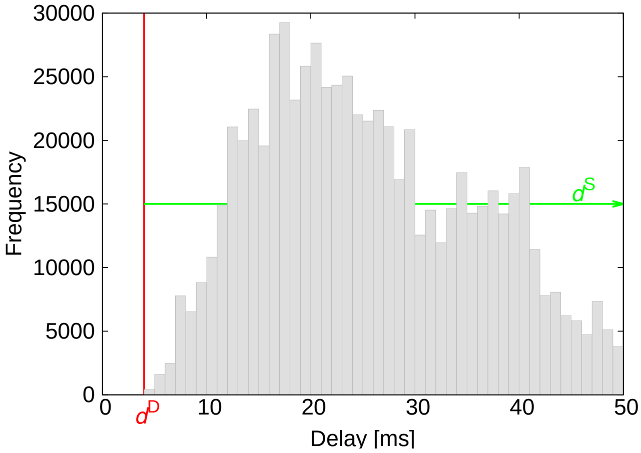


Figure 3: Deterministic and stochastic parts of delay for communication at given distance.

$$d^{\mathrm{D}} = \sum_{i=1}^{N} \left(\frac{p_i}{\mathrm{C}} + d_{i-1}^{\mathrm{RD}} + \frac{M}{b_i} \right) + d^{\mathrm{SD}} + d^{\mathrm{DD}}, \tag{2}$$

where d^{SD} (source deterministic) and d^{DD} (destination deterministic) is delay caused by the generation and reception of the messages at the end hosts. The stochastic part is described as

$$d^{\mathrm{S}}(k) = \sum_{i=1}^{N} d_{i-1}^{\mathrm{RS}}(k) + d^{\mathrm{SS}}(k) + d^{\mathrm{DS}}(k), \tag{3}$$

where $d^{\mathrm{SS}}(k)$ (source stochastic) and d^{DS} (destination stochastic) are delays caused by the actual load of the end hosts. The techniques introduced later work with both parts. The deterministic part is used for deriving geographical constraints as to how far data can travel for a given time from a certain location. The stochastic part is used to derive the most probable distance that data can travel for a given time. The geographical constraints are given by the length of links on the path. Each link introduces the minimum deterministic delay $\frac{p_i}{\mathrm{C}}$. Digital information in optical cables, which is the typical transmission medium in the Internet, is transmitted at the speed of $\frac{2}{3}\mathrm{C} \approx 200$ km/ms [32]. Therefore, the deterministic communication latency for a given distance may be expressed as

$$d^{\mathrm{D}} \geq \sum_{i=1}^{N} \frac{3p_i}{2\mathrm{C}}. \tag{4}$$

The deterministic communication latency also includes other parts. Based on large measurements in the Internet, another constant was found as $\frac{4}{9}\mathrm{C} \approx 133$ km/ms [33] (96th percentile of all measured data). Therefore, the deterministic end-to-end communication latency for a given distance is

$$d^{\mathrm{D}} \approx \frac{9s}{4\mathrm{C}}, \tag{5}$$

where s is the geographical distance between end hosts. This distance is smaller than the sum of the link paths due to circuitous physical layouts and routing policy ($s < \sum_{i=1}^{N} p_i,$). Figure 4 explains the difference. The figure was created on the base of the CESNET backbone network in the Czech Republic [34]. The end hosts (situated in the cities of Cheb and Zlin) are at a specific distance. This distance may be measured as the great-circle distance for the spherical model of the Earth or more accurate geodesic distance for the ellipsoidal model of the Earth. The calculation of these types of distance is covered later. As the red curved lines in the figure show, the actual routing path may be different from the geographically shortest path for reasons explained before (for example, routing policy may prefer higher-bandwidth paths). Therefore, the speed of data communication changes in time due to the actual routing (paths might change in time) and router load. Also, as the figure indicates, there is not a direct path between the location of the routers (in cities), as the cabling is typically installed along major communication paths, such as roads and railways.

The constant $\frac{4}{9}\mathrm{C}$ is not valid over all the ranges of geographical distances. An example of a real measurement of communication latency for a large range of distances (continental and intercontinental) is shown in figure 5. The subfigures plot one-way latency between end hosts. The measurement was again carried out in the PlanetLab network. The red line represents the derived minimum latency for data at a given distance. The line was found as below all the latency/distance plotted points and touching the closest point at the same time (see the CBG method described later for the calculation of such a line).

As shown in the figures, the maximum speed is achieved at various distances. This phenomena is naturally given by the presence of longer optical cables with the transmission speed of $\frac{2}{3}\mathrm{C} \approx 200$ km/ms. There are also differences between intercontinental and continental communication where long submarine optical cables are used. Figure 6a shows the values of higher maximum speed of data transmission between continents for long distances. However, these maximum speeds are rarely achieved. The cumulative distribution functions in figure 6b show the distribution of the actual speed of data transmission globally and for selected continents.

As noted, the measurements presented were carried out in the PlanetLab network [35, 36]. PlanetLab is a large-scale network of Linux servers used for Internet research. These servers

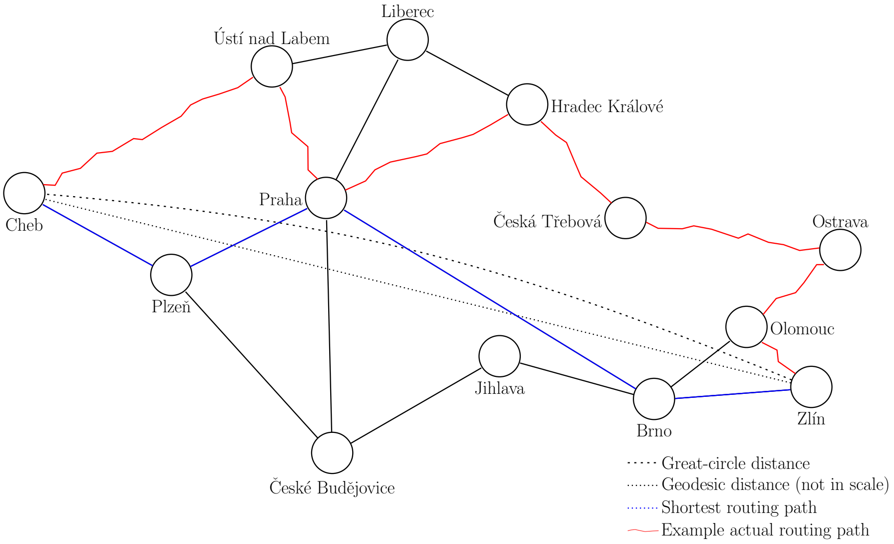


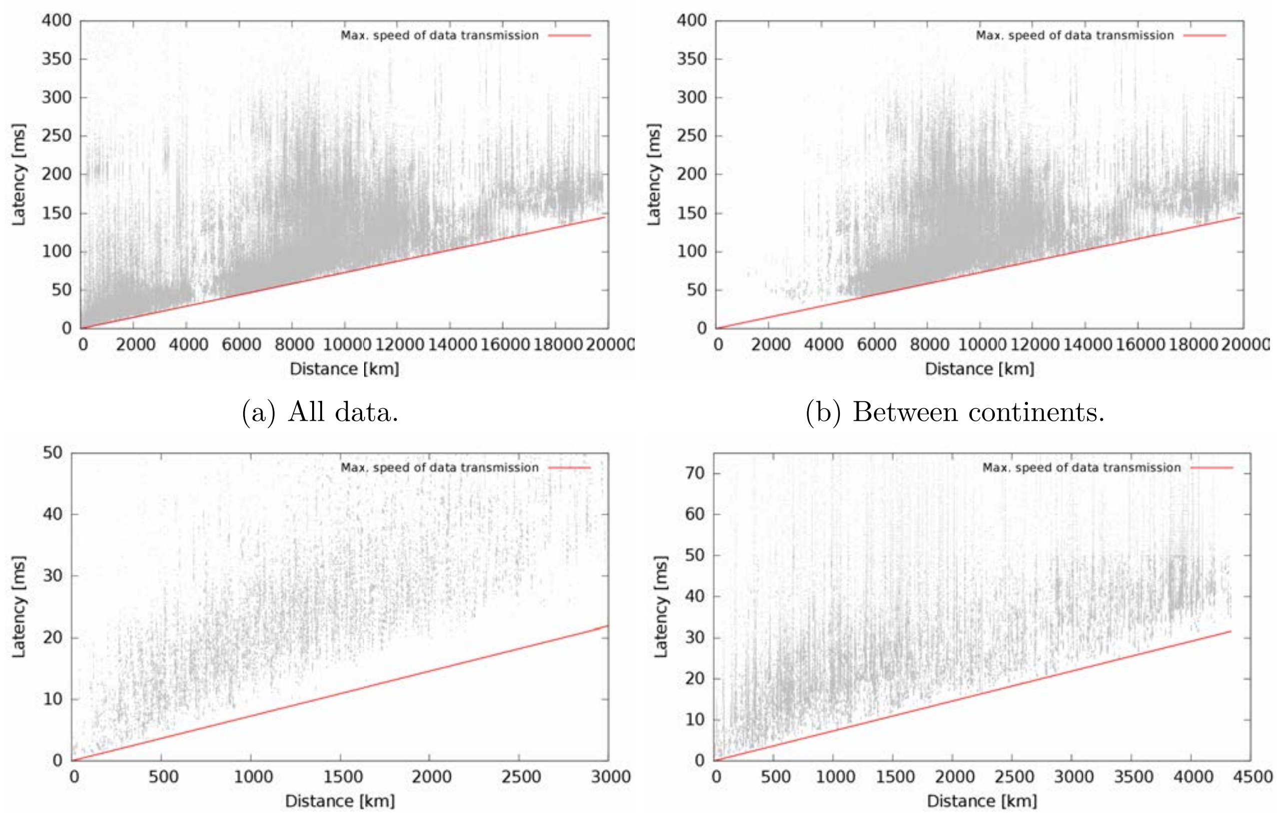
Figure 4: Contributors to deterministic communication latency.

are spread over the whole world and remotely accessible using SSH (Secure Shell). PlanetLab allows engineers and researchers to develop and test their Internet applications. Developing a networking application typically goes through specific steps starting from idea proposal, implementation in a network simulator (such as ns-3, OMNeT++, and NetSim), implementation using a portable code, and testing in a real networking environment. Academics sometimes neglect the last steps and their work has only simulation results. The reason is the lack of easy access to Internet resources. However, the 'wild' world of the Internet cannot be omitted as the simulation results can be different from the true behaviour of the Internet with all the parallel traffic and related problems, such us peak congestion. It is generally difficult to perform such testing as one needs access to a number of Internet nodes to run the developed code and collect the results. This is particularly true when distributed applications, such as cybergeography-related systems, are in question as they typically run on a large number of nodes.

Figure 7b shows the PlanetLab servers clustered according to their location. The two most occupied continents are Europe and North America with more than 50 % of the servers. Users can upload their developed code to the servers and run networking experiments using this code. This is the main difference to the other platforms, such as RIPE Atlas or SamKnows, which are primarily focused on measurements. PlanetLab servers are run by the host organizations (sites) that are typically academic institutions or large technological companies such as Alcatel-Lucent, France Telecom, and Hewlett-Packard. PlanetLab servers run by a host organization share some properties. One of the shared information is their geographical location (that is, the location of the host organization). These locations were used for the distance calculation.

There are virtual machines called slivers run on PlanetLab servers as shown in figure 7a. A sliver is automatically created on a server when a user adds the server to a slice. A slice is a collection of servers a user works with. The server may be selected based on its location to create a geographically distributed system. A user accesses the slivers using a remote SSH connection based on the public/private key authentication associated with a slice.

Several methods may be used to deliver a target location based on known (measured) communication latency. A straightforward approach is based on the observation that a set of hosts is geographically close if they have similar communication latencies to a different set of hosts. This method and the following equations were introduced in [37]. The system consists



(a) All data.

(b) Between continents.

(c) Within Europe.

(d) Within North America.

Figure 5: Estimation of maximum speed of data transmission.

of $\{L_1, ..., L_N\}$ landmarks and $\{M_1, ..., M_J\}$ nodes. The geographical location of the nodes is known – $\mathbf{M} = (M_{\text{lat}}, M_{\text{lon}})$. A node measures latency to a landmark $d_{M,L}$. When latency to all landmarks is known, the jth node constructs a delay vector $MV_j = (d_{j,1}, ..., d_{j,N})$. The target T similarly constructs its delay vector $TV = (d_{T,1}, ..., d_{T,N})$. The Euclidean distance between the delay vectors of a node and the target in N dimensional space is

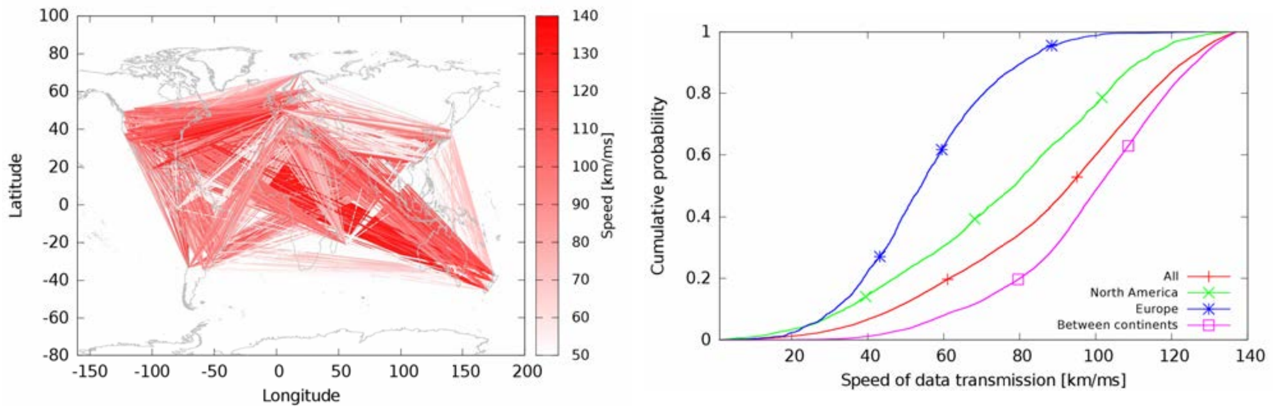
$$dist(TV, MV_j) = \sqrt{\sum_{i=1}^{N} (d_{T,i} - d_{j,i})^2}. \tag{6}$$

The found minimum distance $\min dist(TV, MV_j)$ between target T and jth node is used to define that the location of T is the location of the jth node

$$\arg\min dist(TV, MV_j) \Longrightarrow \mathbf{T} = (T_{\text{lat}}, T_{\text{lon}}) := \mathbf{M}_j = (M_{\text{lat}}, M_{\text{lon}}). \tag{7}$$

Other methods use static or dynamic latency-to-distance conversion to compute the maximum geographical distance between hosts for a given latency. The latency is measured from a set landmarks to the target. The coordinates of a landmark L are known – $\mathbf{L} = (L_{\text{lat}}, L_{\text{lon}})$. The measured latency from a landmark to the target is converted to a maximal distance that defines the radius of a great-circle G_i (on the Earth as the sphere) around landmark L_i. The target is located somewhere within the circle G_i. The intersection of great-circles $\{G_1, ..., G_N\}$ around N landmarks delimits an area R of the target location

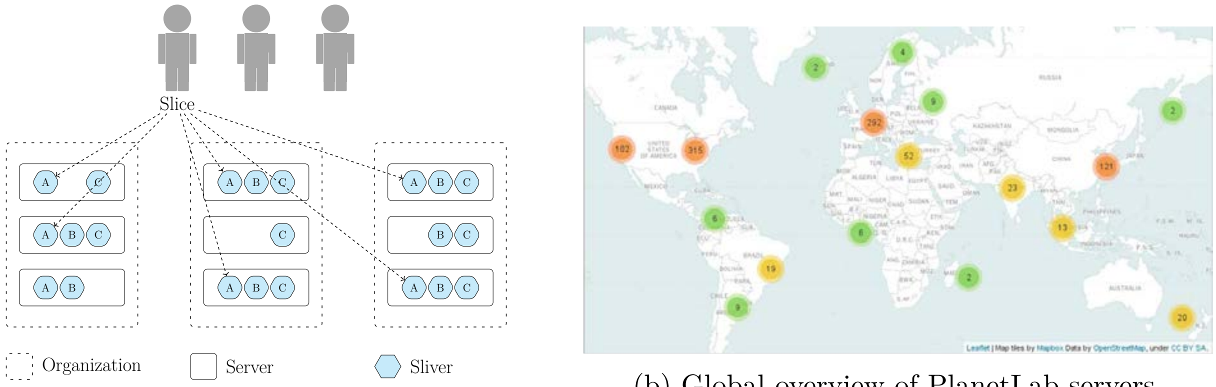
$$R = \bigcap_{i=1}^{N} G_i. \tag{8}$$



(a) Maximum speed of data between continents.

(b) Comparison of actual speed of data.

Figure 6: Typical and maximum speed of data transmission.



Slice
A
B
C
Organization
Server
Sliver
2
4
9
292
102
315
52
2
121
23
6
6
13
19
2
9
20
Leaflet | Map tiles by Mapbox Data by OpenStreetMap, under CC BY SA.

(a) PlanetLab structure.

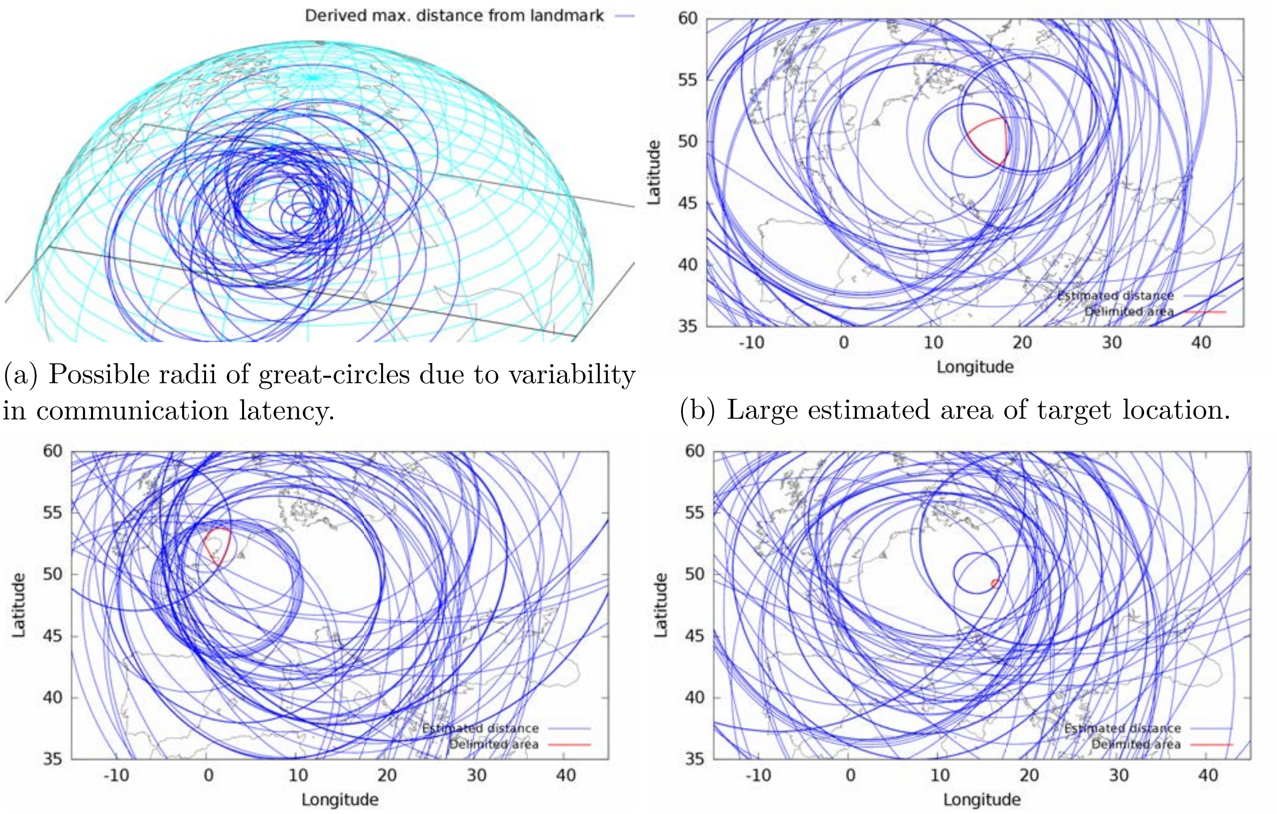
(b) Global overview of PlanetLab servers.

Figure 7: PlanetLab – global experimental network.

Figure 8 shows examples of such an area as a product of great-circle intersections. Figure 8a shows variability of the great-circle radii. The large variability is due to large differences in the latencies measured. These are caused by a number of factors, including the actual load of devices, number of intermediate devices on the path, peak traffic and other possible aspects, such as routing policy. For better accuracy, latency measurements may be periodically repeated to obtain minimum values. Also, as figures 8b, 8c, 8d show, the estimated areas R vary a lot in their sizes. It depends on the application as to determine which maximum area is considered as a usable result. For some cybersecurity applications, areas of confident target location of a country size may be applicable. These applications may be the verification of server authenticity, detection of online identity theft and credit card fraud, and secure routing by avoiding certain regions.

Given the measured latency, the great-circle perimeter is approximated by a polygon formed of N vertexes $\{P_1, ..., P_N\}$. These vertexes $\mathbf{P} = (P_{\text{lat}}, P_{\text{lon}})$ are found at the distance in a chosen azimuth step from the location of landmark $\mathbf{L} = (L_{\text{lat}}, L_{\text{lon}})$. The distance is derived from latency-to-distance conversion. When considering the spherical model of the Earth (shown in figure 8a), the haversine formula [38] calculates the great-circle distance s between a landmark $\mathbf{L}$ and a polygon vertex $\mathbf{P}$ as

$$\text{hav}(\Theta) = \text{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1)\cos(\varphi_2)\text{hav}(\lambda_2 - \lambda_1), \tag{9}$$



(a) Possible radii of great-circles due to variability in communication latency.

(b) Large estimated area of target location.

(c) Medium estimated area of target location.

(d) Small estimated area of target location.

Figure 8: Example area sizes derived by great-circles.

where $\Theta = \frac{s}{r}$ is the central angle between two points on a sphere, r is the radius of the Earth ($\approx 6,371$ km), (φ_1, φ_2) are latitudes of the two points, and (λ_1, λ_2) are their longitudes. The haversine function of the angle θ given by the difference in the points' latitudes and longitudes $(\varphi_2 - \varphi_1, \lambda_2 - \lambda_1)$ is

$$\mathrm{hav}(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1 - \cos(\theta)}{2}. \tag{10}$$

The great-circle distance $s = r \times \mathrm{hav}^{-1}\Big(\mathrm{hav}(\Theta)\Big)$ is solved by the inverse haversine function

$$\mathrm{hav}^{-1}(\theta) = 2\sin^{-1}\left(\sqrt{\theta}\right). \tag{11}$$

Given the coordinates of $\mathbf{L}$ and $\mathbf{P}$, the great-circle distance is calculated as

$$s(\mathbf{L}, \mathbf{P}) = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{L_{\mathrm{lat}} - P_{\mathrm{lat}}}{2}\right) + \cos(P_{\mathrm{lat}})\cos(L_{\mathrm{lat}})\sin^2\left(\frac{L_{\mathrm{lon}} - P_{\mathrm{lon}}}{2}\right)}\right), \tag{12}$$

where $L_{\mathrm{lat}}, P_{\mathrm{lat}}, L_{\mathrm{lon}}, P_{\mathrm{lon}}$ are in radians. The use of the haversine formula for Internet location-aware services was studied in [38]. For more accurate calculations, ellipsoid models of the Earth are used. A common model is WGS 84 [39] with the length of the semi-major axis (equator radius) = 6378137 m, the length of semi-minor axis (poles radius) = 6356752.314245 m, with inverse flattering (1/f) = 298.257223563. Vincenty's formulae straightforwardly solve the direct geodesic problem of finding coordinates of the second point (including reverse azimuth) when the coordinates of the first point, starting azimuth, and distance are given. The following formulae [40] solve the direct problem:

$$\tan\sigma_1 = \frac{\tan U_1}{\cos\alpha_1}, \tag{13}$$

where σ_1 is the angular distance on the auxiliary sphere from equator to $\mathbf{L}$, $U_1 = \arctan\Big((1-f)\tan L_{\text{lat}}\Big)$ is reduced latitude on the sphere, and α_1 is the forward azimuth at $\mathbf{L}$.

$$\sin\alpha = \cos U_1 \sin\alpha_1, \tag{14}$$

where α is the azimuth of the geodesic at the equator.

$$u^2 = \cos^2\alpha \frac{a^2 - b^2}{b^2}, \tag{15}$$

where a is the length of semi-major axis (6378137 m), $b = a(1-f)$ is the length of semi-minor axis of the ellipsoid (6356752.314245 m) with flattering f of the ellipsoid (1/298.257223563), values given for WGS 84.

$$A = 1 + \frac{u^2}{16384}\Big(4096 + u^2\Big(-768 + u^2(320 - 175u^2)\Big)\Big), \tag{16}$$

$$B = \frac{u^2}{1024}\Big(256 + u^2\Big(-128 + u^2(74 - 47u^2)\Big)\Big). \tag{17}$$

The following equations 18, 19, 20 are iterates until there is a little change (for example ≈ 0.006 mm [41]) in σ with initial value of $\frac{s}{bA}$.

$$2\sigma_m = 2\sigma_1 + \sigma, \tag{18}$$

where σ_m is the angular distance on the auxiliary sphere from the equator to the line midpoint and σ_1 is the angular distance on the sphere from the equator to $\mathbf{L}$.

$$\begin{gathered}\Delta\sigma = B\sin\sigma\Big(\cos(2\sigma_m) + \\ \tfrac{B}{4}\Big(\cos\sigma(-1 + 2\cos^2(2\sigma_m)) - \tfrac{B}{6}\cos(2\sigma_m)(-3 + 4\sin^2\sigma)(-3 + 4\cos^2(2\sigma_m))\Big)\Big),\end{gathered} \tag{19}$$

$$\sigma = \frac{s}{bA} + \Delta\sigma. \tag{20}$$

After there is a little change in σ, the direct problem is calculated as

$$P_{\text{lat}} = \arctan\left(\frac{\sin U_1\cos\sigma + \cos U_1\sin\sigma\cos\alpha_1}{(1-f)\sqrt{\sin^2\alpha + (\sin U_1\sin\sigma - \cos U_1\cos\sigma\cos\alpha_1)^2}}\right), \tag{21}$$

$$\lambda = \arctan\left(\frac{\sin\sigma\sin\alpha_1}{\cos U_1\cos\sigma - \sin U_1\sin\sigma\cos\alpha_1}\right), \tag{22}$$

where λ is the difference in longitudes on the auxiliary sphere.

$$C = \frac{f\cos^2\alpha\Big(4 + f(4 - 3\cos^3\alpha)\Big)}{16}. \tag{23}$$

The difference between longitudes E is

$$E = \lambda - (1 - C) f \sin\alpha \Big(\sigma + C \sin\sigma \Big(\cos(2\sigma_m) + C \cos\sigma(-1 + 2\cos^2(2\sigma_m)) \Big) \Big). \tag{24}$$

Finally, the longitude of $\mathbf{P}$ is calculated as

$$P_{\text{lon}} = E + L_{\text{lon}}. \tag{25}$$

The values of $L_{\text{lat}}, L_{\text{lon}}, \alpha_1, P_{\text{lat}}, P_{\text{lon}}$ are in radians. Vincenty's formulae runtime was evaluated in [38] and it was about twice as slow than the haversine formula. The accuracy was validated in [42] with a result of less than a millimetre, even for distances of 18,000 km. This accuracy is far beyond the use needed in Cybergeography.

Given the location of landmark L and using the solution of the direct problem, a closed polygon of K vertexes $\{P_1, ..., P_K\}$, $P_1 = P_{\text{K}}$ is formed for a geodesic distance s from $\mathbf{L}$ (examples are shown in figures 8b, 8c, 8d, the polygon vertexes were obtained for a small azimuth step from L). The polygon intersection area $\{R_1, ..., R_N\}$, $R_1 = R_{\text{N}}$ delimits the target's location. For reference reasons, the target's location T can be set as the centroid C of the delimited polygon as

$$C_x = \frac{1}{6A} \sum_{i=1}^{N} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i),$$
$$C_y = \frac{1}{6A} \sum_{i=1}^{N} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i), \tag{26}$$

where $x = R_{\text{lon}}$, $y = R_{\text{lat}}$ are transformed coordinates to the UTM (Universal Transverse Mercator) 2D Cartesian coordinate system. The target's location is $\mathbf{T} := \mathbf{C}$.

For the above calculations, the geodesic distance from L to polygon vertexes $\{R_1, ..., R_N\}$ need to be known. This distance can be found using the static method of latency-to-distance conversion (Speed of Internet, SOI) that was introduced in [33]. It uses the conversion constant of $\frac{4}{9}$C. An example of dynamic latency-to-distance conversion is the CBG method (Constraint-Based Geolocation) introduced in [43]. The specific latency-to-distance conversion is found for each landmark L. Each landmark measures latency to other landmarks in the system. Between landmark L_i and L_j is the communication latency $d_{i,j}$. The measurement of $d_{i,j}$ may be periodically repeated to reflect the current network performance. The landmarks are at geodesic distance $s_{i,j}$. Each landmark constructs a conversion line that lies under all latency/distance plotted points (x, y) to other landmarks and touches the closest point at the same time. The conversion line for landmark L_i is noted as $y = m_i x + b_i$. For each landmark L_i, the conversion line slope m_i and its intercept b_i

$$y - \frac{d_{i,j} - b_i}{s_{i,j}} x - b_i \geq 0 \tag{27}$$

is to be found as

$$\arg \min_{\substack{b_i > 0 \\ m_i \leq m}} \Bigg(\sum_{\substack{j=1 \\ j \neq i}}^{N} y - \frac{d_{i,j} - b_i}{s_{i,j}} x - b_i \Bigg), \tag{28}$$

where $m_i = \frac{d_{i,j}-b_i}{s_{i,j}}$ is the slope of the conversion line for landmark L_i, m is the slope of the conversion line equal to the speed of $\frac{2}{3}$C, b_i is the latency offset at L_i, and N is the number of landmarks. The geodesic distance $s_{i,T}$ from landmark L_i for target T is

$$s_{i,T} = \frac{d_{i,T} - b_i}{m_i}, \tag{29}$$

where $d_{i,T}$ is the measured latency from L_i to T.

A follow-up approach (Geolocation using buffering delay estimation, GeoBud) [44] uses different latency-to-distance conversion, as paths to targets may have different properties. For each landmark L_i and target T, the delay $d_{\mathrm{i,T}}$ is approximated as

$$d_{\mathrm{i,T}} = m_i \times s_{i,T} + b_{\mathrm{i,T}}, \tag{30}$$

where m_i is the latency-to-distance conversion for data on a path between the landmarks, $s_{i,T}$ is the geodesic distance between landmark L_i and target T, and $b_{\mathrm{i,T}}$ is the total buffering delay on the path from L_i to T. The partial buffering delays b_k are estimated for each hop on the path from L_i to T. The value of $b_{\mathrm{i,T}}$ is the sum of partial delays b_k that are derived from

$$\Delta d_{i,k+1} = d_{i,k+1} - d_{i,k} = m_i \times s_{\mathrm{k,k+1}} + b_{k+1}, \tag{31}$$

where k is the kth router on the path ($k = 0$ is the void router assigned to the first link, $d_{i,0} = 0$), and $s_{\mathrm{k,k+1}}$ is the distance between two routers. The hops and partial delays are obtained from a traceroute measurement. For the calculation of $b_k = \Delta d_{i,k} - m_i \times s_{\text{k-1,k}}$, where $k \neq 0$, the location of each hop needs to be known and can be, for example, derived from a dedicated geolocation database.

Other methods specify the target location area in a different way, for example, as a politically-defined region. Additional input data may also be involved. The method presented in [45] includes hop count and population density in the target area computation. Hop count may be derived using traceroute, as described above, or by the TTL (Time To Live) value in the IP packet header sent by the target, which is a faster method. Different operating systems may set specific initial TTL values that can be used for identification of the number of hops from target to landmark. A summary of the initial values for common operating systems for end hosts and servers are listed in table 7 [46]. The operating system of the target may be detected using TCP/IP fingerprinting [47]. This method also allows the detection of the target device type, which may be useful (along with its geographical location area) in certain cybersecurity applications. The fingerprinting method sends probes to the end host to discover its TCP/IP networking configuration. Some of the considered values include (in brackets are the letters used in fingerprints):

- initial time to live (T),
- maximum segment size (MSS),
- window scale (WS),
- selective ACK permitted (SACK),
- initial window size (W, W1-W6),
- IP do not fragment bit (DF),

Table 7: Selected initial TTL values for ICMP protocol.

Operating system	Version/kernel	TTL
FreeBSD	5	64
Linux	2.4	255
Linux	>2.6	64
OpenBSD	2.6/2.7	255
Windows	7/10	128
MacOS	X	64

- ICMP do not fragment (DFI),
- acknowledgement number (A).

An example fingerprint of a Linux system (CentOS 7 distribution) is shown in listing 7. The test codes and their obtained values are shown in the brackets. Each test is separated by the '%' symbol. The line SCAN gives details about the performed tests, SEQ gives results for sequence generation tests, and T2-T7 is the number of a particular sequent probe (total six probes). The line OPS lists TCP options received for each probe test. The line WIN stands for TCP window sizes for each test. Finally, line IE refers to results obtained from the ICMP echo/request tests. Further information about the format of fingerprints can be found in [48].

```
[]# nmap -O -d 147.229.147.X
TCP/IP fingerprint:
SCAN(V=6.40%E=4%D=8/18%OT=22%CT=8080%CU=%PV=N%G=N%TM=5B77D0DE%P=x86_64-
    redhat-linux-gnu)
SEQ(SP=107%GCD=1%ISR=10C%TI=Z%TS=A)
OPS(O1=M539ST11NW7%O2=M539ST11NW7%O3=M539NNT11NW7%O4=M539ST11NW7%O5=
    M539ST11NW7%O6=M539ST11)
WIN(W1=7120%W2=7120%W3=7120%W4=7120%W5=7120%W6=7120)
...
T2(R=N)
T3(R=N)
T4(R=Y%DF=Y%TG=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
...
IE(R=Y%DFI=N%TG=40%CD=S)
```

Listing 7: TCP/IP fingerprint of Linux, CentOS 7.

The obtained fingerprints are compared to prints stored in a database, which can be found in [49]. This database stores previous fingerprints of known operating systems and devices. A reference fingerprint for CentOS 7 is shown in listing 8. The first three lines include information and timestamps about prints used for this particular reference. The fourth line 'fingerprint' describes the detected operating system.

Another method uses a probabilistic model for latency-to-distance conversion. The most probable region of the target's location is found when region probabilities derived from measurements at all landmarks are combined. The following formulae of the probabilistic latency-to-distance model was introduced in [50] as the Spotter method. The region likelihood to include target T is derived from the random variable $\tau = (\tau_{\text{lat}}, \tau_{\text{lon}})$. The likelihood of the target presence in all the regions on the Earth is determined by a distance probability density function $g_d^L(\tau)$. The conditional probability of $\mathbf{T} \in R$, where R is a defined region, is

```
# Linux 3.10.0-123.13.2.el7.x86_64 #1 SMP Thu Dec 18 14:09:13 UTC 2014
    x86_64 x86_64 x86_64 GNU/Linux (CentOS 7.0)RED
# Linux 3.10.0-229.1.2.el7.x86_64 #1 SMP Fri Mar 27 03:04:26 UTC 2015
    x86_64 x86_64 x86_64 GNU/Linux
# Linux 2.6.32-504.16.2.el6.x86_64 #1 SMP Tue Mar 10 17:01:00 EDT 2015
    x86_64 x86_64 x86_64 GNU/Linux
Fingerprint Linux 2.6.32 or 3.10
...
Class Linux | Linux | 3.X | general purpose
CPE cpe:/o:linux:linux_kernel:3.10
SEQ(SP=EE-104%GCD=1-6%ISR=FC-112%TI=Z%II=I%TS=A)
OPS(O1=M5B4ST11NW7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%O5=
    M5B4ST11NW7%O6=M5B4ST11)
WIN(W1=3890%W2=3890%W3=3890%W4=3890%W5=3890%W6=3890)
...
T3(R=N)
T4(R=N)
T5(R=Y%DF=Y%T=3B-45%TG=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
...
IE(DFI=N%T=3B-45%TG=40%CD=S)
```

Listing 8: Reference fingerprint for Linux, CentOS 7.

$$P(\mathbf{T} \in R | L \bowtie d) = \int_H g_d^L(\tau) d\tau, \tag{32}$$

where $L \bowtie d$ is the condition of delay d between T and L. The distance probability density is isotropic – equal in all directions from the landmark for a given d. Considering a set of landmarks $\{L_1, ..., L_N\}$, each distance probability density $g_{d_i}^{L_i}(\tau)$ is combined to calculate the resulting probability of $\mathbf{T} \in R$ as

$$\begin{aligned} P(\mathbf{T} \in R | L_1 \bowtie d_1, ..., L_N \bowtie d_N) &= P(\mathbf{T} \in R)^{1-N} \prod_{i=1}^{N} P(\mathbf{T} \in R | L_i \bowtie d_i) = \\ & P(\mathbf{T} \in R)^{1-N} \prod_{i=1}^{N} \int_R g_{d_i}^{L_i}(\tau) d\tau, \end{aligned} \tag{33}$$

where unconditional probability $P(\mathbf{T} \in R)$ may employ other possible information about the target's location in R, for example, population density or information from geolocation databases. The authors of [50] further considered these assumptions: i) Distance probability density around a landmark is isotropic. Therefore, $g_d^L(\tau)$ was replaced with a simpler function $f_d^L(\tau)$, which is the distance probability density for L at a given d. ii) $f_d^L(\tau)$ is independent of the location of L and, therefore, it can be further simplified to f_d, which is the same for all the landmarks. To derive f_d, a large-scale measurement in the PlanetLab experimental network was carried out. The delay between PlanetLab servers was analysed with a result that f_d could be approximated with the normal distribution function. The distance distribution function for a delay d is

$$f_d(s) \approx \frac{1}{\sqrt{2\pi\sigma^2(d)}} e^{-\frac{(s-\mu(d))^2}{2\sigma^2(d)}}, \tag{34}$$

where $\mu(d)$ and $\sigma(d)$ are derived from the measurement results for a given d, and s is a random variable describing distance. This distance density function is used to find the probability density $g_d^L(\tau)$ as

$$g_d^L(\tau) = A_d \times f_d\Big(S(L,\tau)\Big), \tag{35}$$

where A_d is the normalization factor, and $S(L,\tau)$ is the distance from L to τ. To calculate the combined probability of the target being in a region, the Earth is divided into a number of regions of a given size. In [50] these regions are spherical triangles of a similar size obtained from Hierarchical Triangular Mesh [51]. For the selected size of the triangles, the combined distance probability density is calculated by equation (33). The final region of the target's location can be derived as a union of a set of smaller triangles with the highest calculated probabilities. If a reference location is needed, it can be derived as the centroid (eq. 26) of the most probable triangle after transformation to an appropriate coordinate system.

5 SELECTED SECURITY APPLICATIONS

There is a variety of location-aware Internet applications based on knowledge in Cybergeography. These include content personalization, marketing, digital rights management, and user behaviour analysis. The selected cybersecurity-related applications are the verification of server authenticity, detection of online identity theft, detection of credit card fraud, and secure routing by avoiding certain regions.

- *Verification of server authenticity.* Known information about the server location in a reasonably sized region may be included in its authentication procedure [6]. An example is shown in figure 9 where four landmarks (with known location) verify the web server location for a client connecting to the server. The landmarks measure communication latency to the web server reported by the client when accessing its website. If the actual server is found within the previously verified region for the genuine server, the client is informed about the successful geographical authentication. Including this kind of verification in the authentication process reduces the risk of connecting to a fake website, which may lead to stealing user's personal information (authentication credentials). An overview of this kind of attack is given in table 8 [6]. A fake website may be provided to the user by a phishing (deceptive URL is provided, for example, in an email) or pharming attack (altered domain name of the web server is stored in a local DNS cache, the 'hosts' file or at any level of the global DNS system). Other possible forms of the fake-website attack may be applied topologically closer to the genuine web server, for example in its LAN (Local Area Network). In this case, such an attack has global effect as it affects all the connecting users. An example is ARP (Address Resolution Protocol) spoofing in the LAN of the genuine web server. This form of attack overrides the MAC (Media Access Control) address of the genuine web server in the gateway's IP-to-MAC table and thus redirects all users to a fake web server. In this case, the web server's domain and IP address are unchanged. Another form of attack may happen at the link layer. A switch in the LAN consults its MAC-to-port table to identify the outcoming port where to forward frames intended for the web server. By modifying the switch's MAC table, the traffic may be forwarded to a fake web server. In this case, server domain, IP address and even its MAC address is unchanged. The presence of a fake web server in the LAN of the genuine server is unlikely. Using a compromised device in the LAN, the traffic may be re-routed to a fake server at a different location controlled by the attacker. This re-routing introduces additional latency that alters the latency patterns leading to region verification failure. An option is that the fake server may relay the traffic to the original server and thus implement the man-in-the-middle attack. Another application is to prevent access to servers in suspicious regions, for example identified from the previous attacks.

Table 8: Characteristic of fake website attacks.

How	Effect	Device attacked	Changed identifiers
Phishing	Local	Host	Domain, IP, MAC, Switch port
Pharming	Moderate	Host, DNS server	IP, MAC, Switch port
ARP spoofing	Global	LAN host	MAC, Switch port
MAC table	Global	LAN switch	Switch port

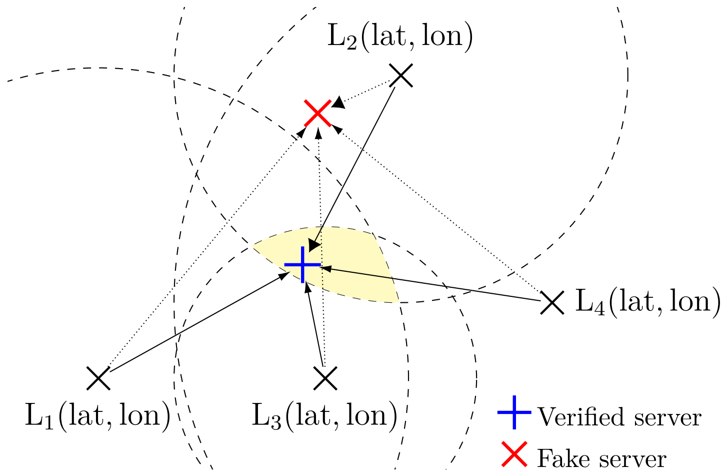


Figure 9: Web server verification by its location region.

- *Detection of online identity theft and credit card fraud.* Impersonation of a user's identity by using their credentials when accessing a service could lead to various offences including bullying, blackmailing, stealing, and general credit misuse. The early detection of identity theft may prevent these offences. Evaluation of the spatial location of a user's device is one of the methods used. The known location of the user's device at the instant of accessing a service is evaluated in various ways, such as the change to a new, not previously visited place, or the time between consequent logins from distant places [4, 52]. A threshold may be set for user travel velocity to detect logins that are not possible for a single person. Such a threshold could be as much as 800 km/h [53], which is the general speed of commercial flights. Different velocity thresholds are used, for example, along with other indices of fraud, such as rapid changes in user behaviour (increased frequency of logins/transactions or amounts involved in transactions). The velocity threshold may also be set according to the user profile derived from their travel history (i.e. history of transaction locations) and for different countries depending on their size. For example, if both transaction locations are not within the same country, a lower threshold of 400 km/h can be set [53]. Other information for setting a profiled velocity threshold could be the user device type (a method for device type detection was discussed in section 4). For example, if the same mobile device is used for successive credit card transactions, a higher velocity threshold is set. For desktop computers, a lower velocity threshold is used. Credit card fraud is related to identity theft with a difference of the data impersonated. The geographical location and time is recorded for online transactions issued from a device. Similar to the previous scenario, the velocity between transaction locations is evaluated. If the velocity is above the threshold, the credit card may be rejected by the bank that issued the card or the card payment system as being impersonated.

- *Secure routing by avoiding certain regions.* By knowing the spatial location of the routers on the path between end hosts, certain regions may be avoided by altering the routing policy [7]. These regions of avoidance may include countries with a high risk of eavesdropping, or other factors may be considered, such as censorship. Avoiding specific regions may be a complementary technique assuring a higher security of sensitive data transmission. Even the knowledge of encrypted data being transmitted and their volume may provide some information leading to a security breach. Other risk of transmitting encrypted data over untrusted regions is some traffic being intentionally dropped to disrupt the communication. Along with the location of the routers on the path, other information may be employed in the routing policy of avoidance, such as the router operating system version to avoid untrusted devices with known vulnerability issues [7]. A method for detecting operating systems was described in section 4.

REFERENCES

[1] CHIARA, D. et al. Geomarketing Policies and Augmented Reality for Advertisement Delivery on Mobile Devices. In: *Proceedings of the 17th International Conference on Distributed Multimedia Systems.* Knowledge Systems Institute, 2011, pp. 78–83.

[2] BARNES, R. et al. *An Architecture for Location and Location Privacy in Internet Applications.* IETF, 2011. Request for Comments: 6280.

[3] YEN, N. et al. Intelligent route generation: discovery and search of correlation between shared resources. *International Journal of Communication Systems.* 2013, vol. 26, no. 6, pp. 732–746.

[4] AIMEUR, E.; SCHONFELD, D. The ultimate invasion of privacy: Identity theft. In: *Ninth Annual International Conference on Privacy, Security and Trust.* IEEE, 2011, pp. 1–8.

[5] GULATI, A. et al. Credit card fraud detection using neural network and geolocation. *IOP Conference Series: Materials Science and Engineering.* 2017, vol. 263, no. 4, pp. 1–6.

[6] ABDOU, A.; OORSCHOT, P. Server Location Verification (SLV) and Server Location Pinning: Augmenting TLS Authentication. *ACM Transactions on Privacy and Security.* 2018, vol. 21, no. 1, pp. 1–26.

[7] KLINE, E.; REIHER, P. Securing Data Through Avoidance Routing. In: *Proceedings of the 2009 workshop on New security paradigms.* ACM, 2009, pp. 115–124.

[8] HUFFAKER, B.; FOMENKOV, M.; CLAFFY, K. *Geocompare: a comparison of public and commercial geolocation databases.* CAIDA, 2011. Technical Report, University of California.

[9] MORAVEK, P. et al. Study and performance of localization methods in IP based networks: Vivaldi algorithm. *Journal of Network and Computer Applications.* 2011, vol. 34, no. 1, pp. 351–367.

[10] GUO, C. et al. Mining the Web and the Internet for Accurate IP Address Geolocations. In: *IEEE International Conference on Computer Communications 2009.* IEEE, 2009, pp. 2841–2845.

[11] FIOREZE, T.; HEIJENK, G. Extending DNS to support geocasting towards VANETs: A proposal. In: *Vehicular Networking Conference (VNC) 2010.* IEEE, 2010, pp. 271–277.

[12] MENS, J. *Where is your DNS server LOCated?* [online]. 2010. [accessed August 2018]. Available: `jpmens.net/2010/11/14/where-is-your-dns-server-located/`.

[13] MORAVEK, P. et al. Investigation of radio channel uncertainty in distance estimation in wireless sensor networks. *Telecommunication systems.* 2013, vol. 52, no. 3, pp. 1549–1558.

[14] ITU-T. *A Study on the IPv6 Address Allocation and Distribution Methods.* University Sains Malaysia, 2009. Document 3-PLEN.

[15] KOMOSNY, D.; VOZNAK, M.; REHMAN, S. Location Accuracy of Commercial IP Address Geolocation Databases. *Information Technology And Control.* 2017, vol. 46, no. 3.

[16] BALAKRISHNAN, H. et al. A layered naming architecture for the internet. In: *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications.* ACM, 2004, pp. 343–352.

[17] POESE, I. et al. IP geolocation databases: unreliable? *ACM SIGCOMM Computer Communication Review.* 2011, vol. 41, no. 2, pp. 53–56.

[18] FIOREZE, T.; HEIJENK, G. Extending the Domain Name System (DNS) to provide geographical addressing towards vehicular ad-hoc networks (VANETs). In: *Vehicular Networking Conference (VNC) 2011.* IEEE, 2011, pp. 70–77.

[19] SOUSA, A. et al. Using DNS to establish a Localization Service. In: *International Conference on Indoor Positioning and Indoor Navigation 2014.* IEEE, 2014, pp. 385–392.

[20] MAXMIND. *GeoLite2 Free Downloadable Databases* [online]. 2018. [accessed August 2018]. Available: `dev.maxmind.com/geoip/geoip2/geolite2/`.

[21] MAXMIND. *GeoIP2 Precision Services* [online]. 2018. [accessed August 2018]. Available: `www.maxmind.com/en/geoip2-precision-services`.

[22] SKYHOOK WIRELESS. *Hyperlocal IP* [online]. 2018. [accessed August 2018]. Available: `resources.skyhookwireless.com/wiki/type/documentation/hyperlocal-ip/`.

[23] DIGITAL ENVOY. *NetAcuity Product Sheet* [online]. 2018. [accessed August 2018]. Available: `www.digitalelement.com/resources/data-sheets/`.

[24] MAXMIND. *GeoIP2 City Accuracy* [online]. 2016. [accessed March 2016]. Available: `www.maxmind.com/en/geoip2-city-accuracy-comparison`.

[25] IP2LOCATION.COM. *IP2Location Data Accuracy* [online]. 2016. [accessed March 2016]. Available: `www.ip2location.com/data-accuracy`.

[26] NEUSTAR. *Accuracy of Neustar's IP Intelligence Services* [online]. 2016. [accessed March 2016]. Available: `www.security.neustar/digital-performance/web-performance-management`.

[27] GEOBYTES. *Geobytes Frequently Asked Questions* [online]. 2016. [accessed March 2016]. Available: `geobytes.com/faq/`.

[28] SHAVITT, Y.; ZILBERMAN, N. A Geolocation Databases Study. *IEEE Journal on Selected Areas in Communications*. 2011, vol. 2, no. 10, pp. 2044–2056.

[29] HAN, K. et al. Internet Control Architecture for Internet-Based Personal Robot. *Autonomous Robots*. 2001, vol. 10, no. 2, pp. 135–147.

[30] YANG, S. et al. Time delay and data loss compensation for Internet-based process control systems. *Transactions of the Institute of Measurement and Control*. 2005, vol. 27, no. 2, pp. 103–118.

[31] BOVY, C. et al. Analysis of end-to-end delay measurements in Internet. In: *Passive and Active Measurement Workshop-PAM 2002*. Agilent Laboratories, 2002, pp. 1–8.

[32] PERCACCI, R.; VESPIGNANI, A. Scale-free behavior of the Internet global performance. *The European Physical Journal B - Condensed Matter and Complex Systems*. 2003, vol. 32, no. 4, pp. 411–414.

[33] KATZ-BASSETT, E. et al. Towards IP geolocation using delay and topology measurements. In: *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*. ACM, 2006, pp. 71–84.

[34] SLAVICEK, K.; NOVAK, V.; LEDVINKA, J. CESNET Fiber Optics Transport Network. In: *Eighth International Conference on Networks*. IEEE, 2009, pp. 403–408.

[35] CHUN, B. et al. PlanetLab: an overlay testbed for broad-coverage services. *ACM SIGCOMM Computer Communication Review*. 2003, vol. 33, no. 3, pp. 3–12.

[36] KOMOSNY, D. et al. Testing Internet applications and services using PlanetLab. *Computer Standards & Interfaces*. 2017, vol. 53, pp. 33–38.

[37] PADMANABHAN, V.; SUBRAMANIAN, L. An investigation of geographic mapping techniques for internet hosts. *ACM SIGCOMM Computer Communication Review*. 2001, vol. 31, no. 4, pp. 173–185.

[38] MAHMOUD, H.; AKKARI, N. Shortest Path Calculation: A Comparative Study for Location-Based Recommender System. In: *World Symposium on Computer Applications & Research (WSCAR)*. IEEE, 2016, pp. 1–5.

[39] SLATER, J.; MALYS, S. WGS 84 - Past, Present and Future. In: *Advances in Positioning and Reference Frames*. Springer, 1998, pp. 1–7.

[40] VINCENTY, T. Direct and Inverse Solutions of Geodesics on the Ellipsoid with application of nested equations. *Survey Review*. 1975, vol. XXIII, no. 176, pp. 88–93.

[41] VENESS, C. *Vincenty solutions of geodesics on the ellipsoid* [online]. 2016. [accessed August 2018]. Available: `www.movable-type.co.uk/scripts/latlong-vincenty.html`.

[42] THOMAS, C.; FEATHERSTONE, W. Validation of Vincenty's Formulas for the Geodesic Using a New Fourth-Order Extension of Kivioja's Formula. *Journal of Surveying Engineering*. 2005, vol. 131, no. 1, pp. 20–26.

[43] GUEYE, B. et al. Constraint-based geolocation of internet hosts. *IEEE/ACM Transactions on Networking*. 2006, vol. 14, no. 6, pp. 1219–1232.

[44] GUEYE, B. et al. Leveraging Buffering Delay Estimation for Geolocation of Internet Hosts. In: *The 5th International IFIP-TC6 Conference on Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*. Springer, 2006, pp. 319–330.

[45] ERIKSSON, B. et al. A learning-based approach for IP geolocation. In: *Proceedings of the 11th international conference on Passive and active measurement*. Springer, 2010, pp. 171–180.

[46] SIBY, S. *Default TTL (Time To Live) Values of Different OS* [online]. 2014. [accessed August 2018]. Available: `subinsb.com/default-device-ttl-values/`.

[47] GORDON, L. et al. *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Insecure.Com LLC, 2009. ISBN 978-0979958717.

[48] GORDON, L. *Understanding an Nmap Fingerprint* [online]. Insecure.Com LLC, 2011. [accessed August 2018]. Available: `nmap.org/book/osdetect-fingerprint-format.html`.

[49] INSECURE.COM LLC. *Nmap OS Fingerprinting 2nd Generation DB* [online]. Community contributors, 2017. [accessed August 2018]. Available: `svn.nmap.org/nmap/nmap-os-db`.

[50] LAKI, S. et al. Spotter: A model based active geolocation service. In: *IEEE Conference on Computer Communications*. IEEE, 2011, pp. 3173–3181.

[51] GYORGY, F. et al. *Hierarchical Triangular Mesh* [online]. 2007. [accessed August 2018]. Available: `www.skyserver.org/HTM/`.

[52] CHO, Y.; LEE, S. Detection and Response of Identity Theft within a Company Utilizing Location Information. In: *International Conference on Platform Technology and Service (PlatCon)*. IEEE, 2016, pp. 1–5.

[53] BUCHHOP, P. *Use of Velocity in Fraud Detection or Prevention*. 2013. Patent no. US20130110715A1, Assignee: Bank of America Corp.

ABSTRAKT

Přednáška se zabývá současným stavem poznání v interdisciplinární oblasti kybergeografie. Prezentované přístupy popisují možnosti zjištění geografické oblasti, ve které se nachází zařízení připojené do sítě Internet. Jedná se o obecné přístupy pracující vzdáleně, které lze aplikovat na jakékoliv zařízení v Internetu bez znalosti jeho nasazení (mobilní, pevné), dostupné hardwarové výbavy (není použito systému GPS, WiFi triangulace, atd.) a instalovaných aplikací. Je představen způsob přidělování adresního prostoru a organizace speciálních geolokačních databází. Rozšířený popis je věnován analýze komunikace v síti Internet, jelikož zde probíhá vývoj v aplikační oblasti. Představeny jsou metody založené na vytýčení hraničních oblastí a pravděpodobnostním modelování. Znalosti v oboru kybergeografie jsou základem pro realizaci velké škály aplikací, které pracují s polohou obecných zařízení. Například se jedná o personalizaci webových stránek a analýzu chování uživatelů. Přednáška zahrnuje vybrané aplikace v oblasti kyberbezpečnosti, kterými jsou detekce zcizení on-line identity, detekce zneužití kreditní karty a ověření autenticity serveru.

ABSTRACT

The lecture deals with the current knowledge in the interdisciplinary field of Cybergeography. The presented approaches describe methods for delimiting a geographical area that includes an Internet device. The general remote methods are considered; these may be used for any Internet device without knowledge of its use (mobile, fixed), available hardware resources (GPS and WiFi triangulation is not considered), and installed applications. A description of address allocation and dedicated geolocation databases is presented. The lecture goes into more detail in analysing network communication since new applications emerge in this area. The measurement-based methods include constraint-based geolocation and probabilistic modelling. Knowledge in Cybergeography is used for implementing a vast number of location-aware applications, including web content personalization and user behaviour analysis. The lecture covers cybersecurity applications dealing with the detection of on-line identity theft and credit card fraud, and verification of server authenticity.